

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2022/2023 (TERM ID 2220)

DCP5201 – OBJECT ORIENTED PROGRAMMING

(All sections / Groups)

05 JULY 2023
9.00 a.m – 11.00 a.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question Paper consists of 17 pages including cover page.
2. There are 3 sections : A,B,C. Answer ALL questions. The distribution of the marks for each question is given. Total Marks is 60.
3. Please write all your answers in this question paper at the space provided.

SECTION A : TRUE or FALSE [10 Marks]

Write True or False for the following statements.

1. In object-oriented programming, the object encapsulates both the data and functions that operate on the data.
Answer : _____

2. A technique known as a member initialization can be used to initialize data members of a class.
Answer : _____

3. A private member function is useful for tasks that are internal to the class and it can also be directly called by statements outside the class.
Answer : _____

4. While a class's member functions may be overloaded, the constructor cannot be overloaded.
Answer : _____

5. When a class declares an entire class as its friend, the friendship status is reciprocal. That is, each class's member functions have free access to the other's private members.
Answer : _____

6. When you overload an operator, you can change the operator's original meaning to something entirely different.
Answer : _____

7. By default, when an object is assigned to another object, each member of one object is copied to its counterpart in the other object.
Answer : _____

8. More than one class may be derived from a base class.
Answer : _____

9. A derived class may become a base class if another class is derived from it.
Answer : _____

10. Pointers to a base class cannot be assigned the address of a derived class object.
Answer : _____

Continued

SECTION B : SHORT ANSWER QUESTIONS [24 Marks]

There are three (3) questions. Answer ALL questions.

QUESTION 1 [6 Marks]

The following program finds the largest area among 3 boxes stored in an array of size 3 with the initialization value for the width and height given in the array declaration. However, the code has errors. Find and correct the error. (6 Marks)

```
1  #include(iostream)
2  using namespace std;
3  class Box{
4      private :
5          float width, height, area;
6      public :
7          Box(float w, float h): w(width), h(height)
8          {   }
9
10         float friend findLargest(Box []);
11
12         void calculate()
13         {   area = width * height;   }
14
15         ~Box()
16         {   cout<<"End of area calculation."<<endl; }
17 };
18
19 float findLargest(Box B)
20 {
21     float largest=0.0;
22     for(int i=0;i<3;i++)
23     {
24         if(B[i].area>largest)
25             largest = B[i].area;
26     }
27     return largest;
28 }
29
30 int main()
31 {
32     Box A[3]={A(3,5), A(4,7), A(2,8)};
33     for(int i=0;i<3;i++)
34         calculate(A[i]);
35     cout<<"Largest box area : "<<A.findLargest()<<endl;
36     return 0;
37 }
```

Continued

Answer :

Line No	Correction
Example :1	<code>#include<iostream></code>

Continued

QUESTION 2 [8 Marks]

Given the following class declaration, complete the main() function at part labelled Y. In main(), create a dynamic memory allocation for an array with the size of value *number* input by the user. Access the class's member functions using arrow operator (->) and indirection (*) dot(.) operator. (8 Marks)

```
#include<iostream>
#include<iomanip>
using namespace std;

class Phone{
    private :
        string brand;
        float price;
    public :
        void setData(string b, float p)
        {
            brand = b;
            price = p;
        }

        string getBrand()
        {
            return brand;
        }

        float getPrice()
        {
            return price;
        }
};

int main()
{
    Phone *ptr;
    int number;
    string phoneBrand;
    float phonePrice;

    cout<<"Enter number of brand : ";
    cin>>number;

    Y

}
```

Continued

Sample Output : [Bold words are input by user]

Enter number of brand : 3

Enter brand : **Oppa**

Enter price : **1099.00**

Enter brand : **Zamzung**

Enter price : **1899.00**

Enter brand : **Xixo**

Enter price : **1499.00**

Brand	Price (RM)
Oppa	1099.00
Zamzung	1899.00
Xixo	1499.00

Answer :

Continued

QUESTION 3 [10 Marks]

a) Referring to the following incomplete program and sample output, define member function that overloads the '+' addition operator at the segment labelled X. (3 Marks)

```
#include<iostream>
#include<iomanip>
using namespace std;

class Invoice
{
    private:
        float total;
    public:
        Invoice(float t)
        {
            total = t;
        }

        X

        void print()
        {
            cout<<fixed<<setprecision(2)<<total<<endl;
        }
};

int main()
{
    Invoice inv1(1500.00), inv2(1200.50);
    Invoice inv3 = inv1 + inv2;
    cout<<"Total invoice : ";
    inv3.print();
    return 0;
}
```

Sample output :

```
Total invoice : 2700.50
```

Answer :

Continued

b) Write the output of the following program.

(4 Marks)

```
class Employee
{
    protected:
        float salary;

    public:
        virtual void display (float s)
        {
            salary = s;
            cout<< "Salary is : RM " <<salary<< endl;
        }

        virtual ~Employee()
        {
            cout<<"End of display."<<endl<<endl;
        }
};

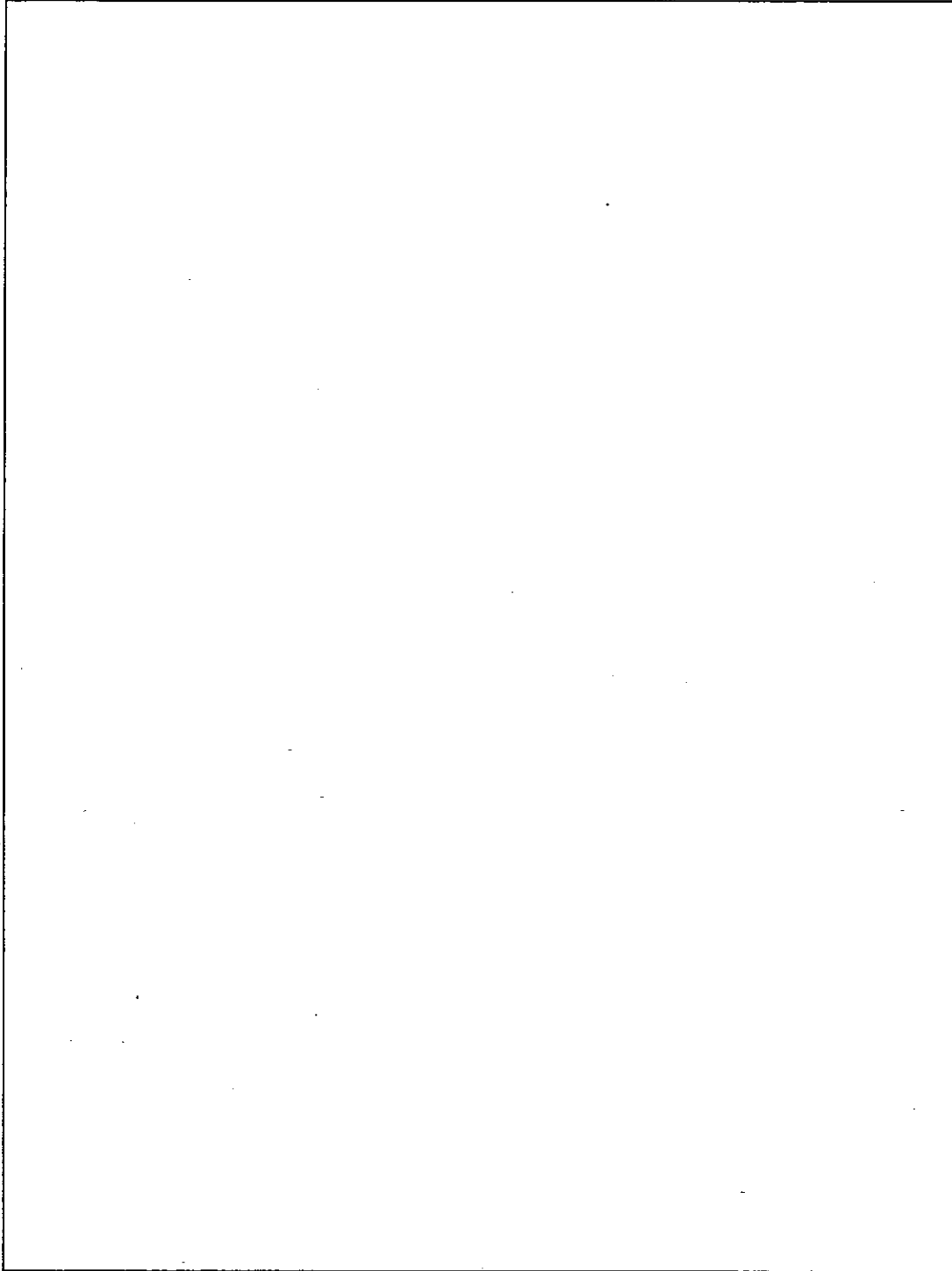
class Manager:public Employee
{
    void display(float s)
    {
        salary = s;
        cout<<"This is a manager's salary : RM "<<salary<< endl;
    }

    ~Manager()
    {
        static int num=1;
        cout << "Manager " <<num<<" salary displayed."<<endl ;
        num++;
    }
};

int main()
{
    Employee E;
    Employee *emp = new Manager;
    emp->display(5000);
    delete emp;
    emp = new Manager;
    emp->display(6000);
    delete emp;
    emp = &E;
    emp->display(7000);
    return 0;
}
```

Continued

Answer :



Continued

c) Given the program shown below, trace the output.

(3 Marks)

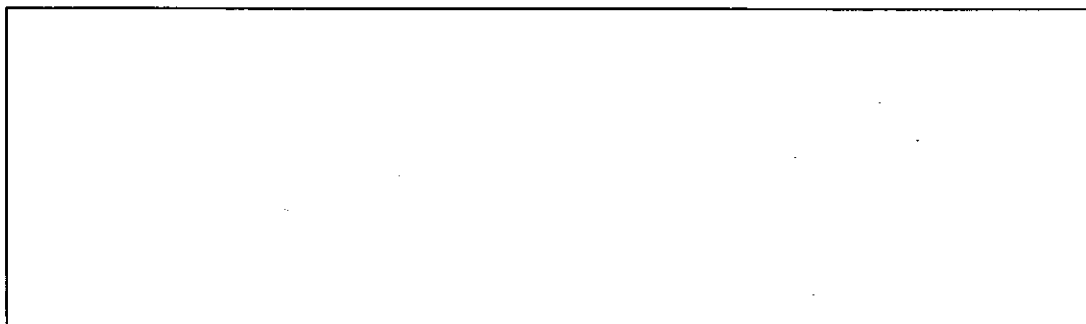
```
class Triangle
{
    protected:
        double a, b, c;
    public:
        Triangle(double x, double y)
        {
            a=x; b=y;
            cout<<"Triangle 2"<<endl;
        }
};

class Pythagoras : protected Triangle
{
    public:
        Pythagoras(double x, double y) : Triangle(x,y)
        {
            cout<<"Pythagoras"<<endl;
        }

        double find()
        {
            c = sqrt(a*a + b*b);
            return c;
        }
};

int main()
{
    double side1=3, side2=4;
    Pythagoras P(side1,side2);
    cout<<"Hypertenuse : "<<P.find()<<endl;
    return 0;
}
```

Answer :



Continued

SECTION C : PROGRAM WRITING QUESTION [26 Marks]

Write a complete program that gets user input for the sales of three luxurious watch for the first quarter of the year. The program will display a summary showing the total and average sales for each watch.

Refer to sample output given below. The **bold items** are inputs entered by user.

- i) Declare a constant named *SIZE* (int). Set the constant value to 4.
- ii) Create a class called *Watch*:
 - Protected data member :
 - *type* : string
 - *material* : string
 - Public member functions:
 - *setWatch(...)* : It is a mutator function that contains two string parameters to set the *type* and *material*.
 - *setSales()* : Set it as a pure virtual function.
- iii) Create class called *DesignerWatch* [derived publicly from class *Watch*]:
 - Private data members
 - *name* : string
 - *month* : string[SIZE]
 - *sales* : float[SIZE]
 - *total* : float
 - Public member functions:
 - *Parameterized constructor* :
 - Contains three parameters of string type. One of the parameter is used to set the *name* and the other two parameters will be passed to function *setWatch(...)* to set the *type* and *material*.
 - Set the array *month* with values "Jan", "Feb", "Mac", "Apr".
 - Set the *total* to 0.
 - *setSales()*:
 - Get user input for *sales* array.
 - Accumulate the *sales* array elements and stored the total value in data member *total*.
 - *display()* :
 - Display the *name*, *type*, *material* and *total*.
 - *Average(DesignerWatch &)*:
 - Set this function as a *friend* of the class. Refer to the instructions after this to define the function.

Continued

- iv) Create a friend function called *Average(DesignerWatch &)*.
- Parameter : Reference object of *DesignerWatch* class.
 - Returns the average sales to *main()* function. Use value of *total* (data member of *DesignerWatch* class) that has been calculated in *setSales()* to calculate the average.
- v) In *main()*:
- Declare a pointer object, *p* of *DesignerWatch* class.
 - Use this pointer to create a dynamic object element. Pass the values "Chopart", "mechanical", "gold" that will set the object's *name*, *type* and *material*.
 - Call *setSales()*.
 - Declare a pointer object, *q* of *DesignerWatch* class.
 - Use this pointer to create a dynamic object element. Pass the values "Rolez Daytona", "mechanical", "oystersteel" that will set the object's *name*, *type* and *material*.
 - Call *setSales()*.
 - For each dynamic objects *p* and *q* :
 - Call *display()*
 - Then, display the average of the sales by calling friend function *Average(...)*, passing the dynamic object as parameter.
 - Deallocate memory for all the dynamic objects.

Sample Output : [Bold words are input by user]

```
***Sales Data Entry for Chopart 201***
Jan sales: 150000.00
Feb sales: 186000.00
Mac sales: 276000.00
Apr sales: 254000.00

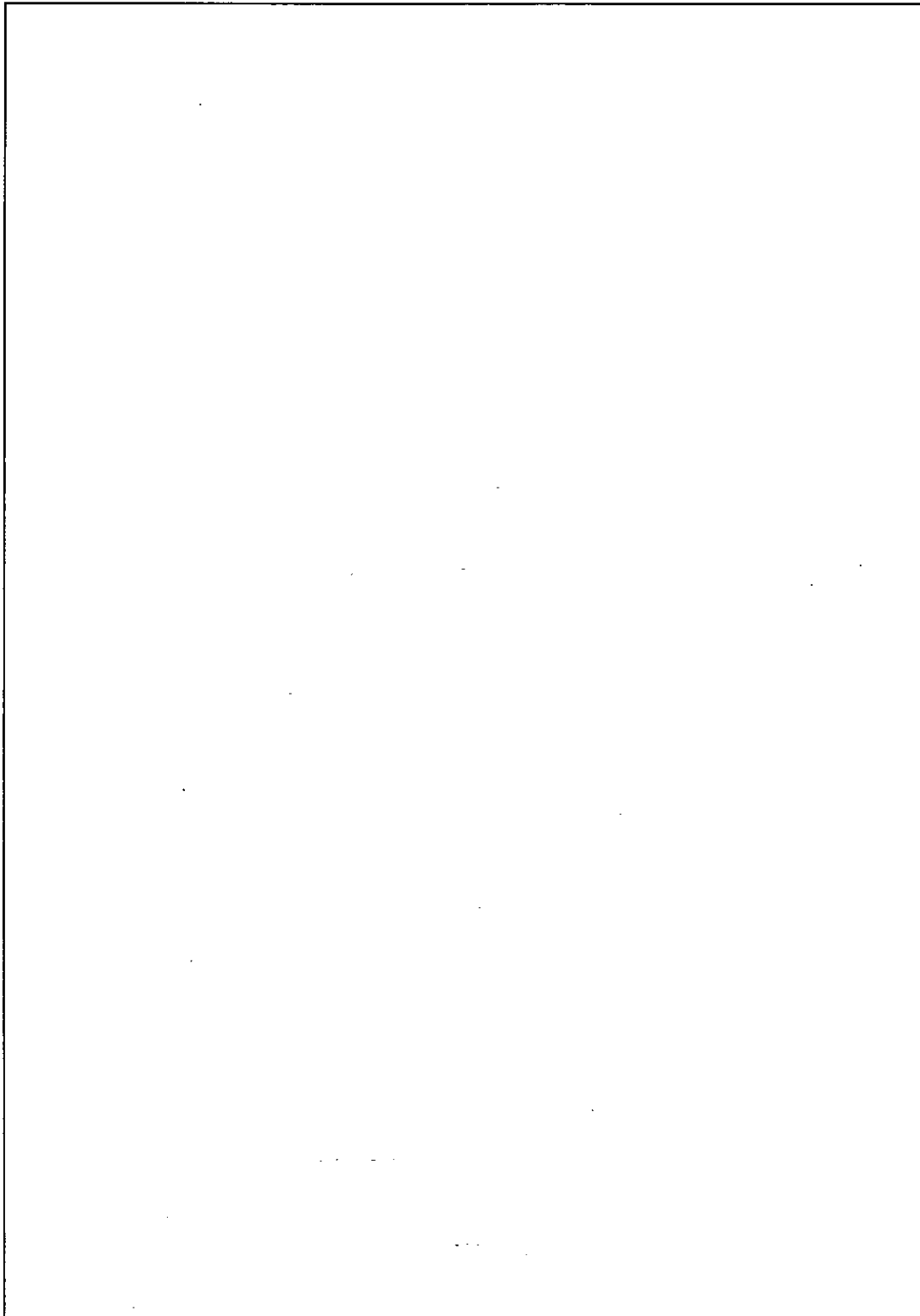
***Sales Data Entry for Rolez Daytona***
Jan sales: 120876.00
Feb sales: 270000.00
Mac sales: 465000.00
Apr sales: 357000.00

-----Summary Report-----
      Name      Type      Material  Total Sales (RM)  Average (RM)
Chopart 201    mechanical      gold      866000.00      216500.00
Rolez Daytona  mechanical  oystersteel  1212876.00      303219.00
```

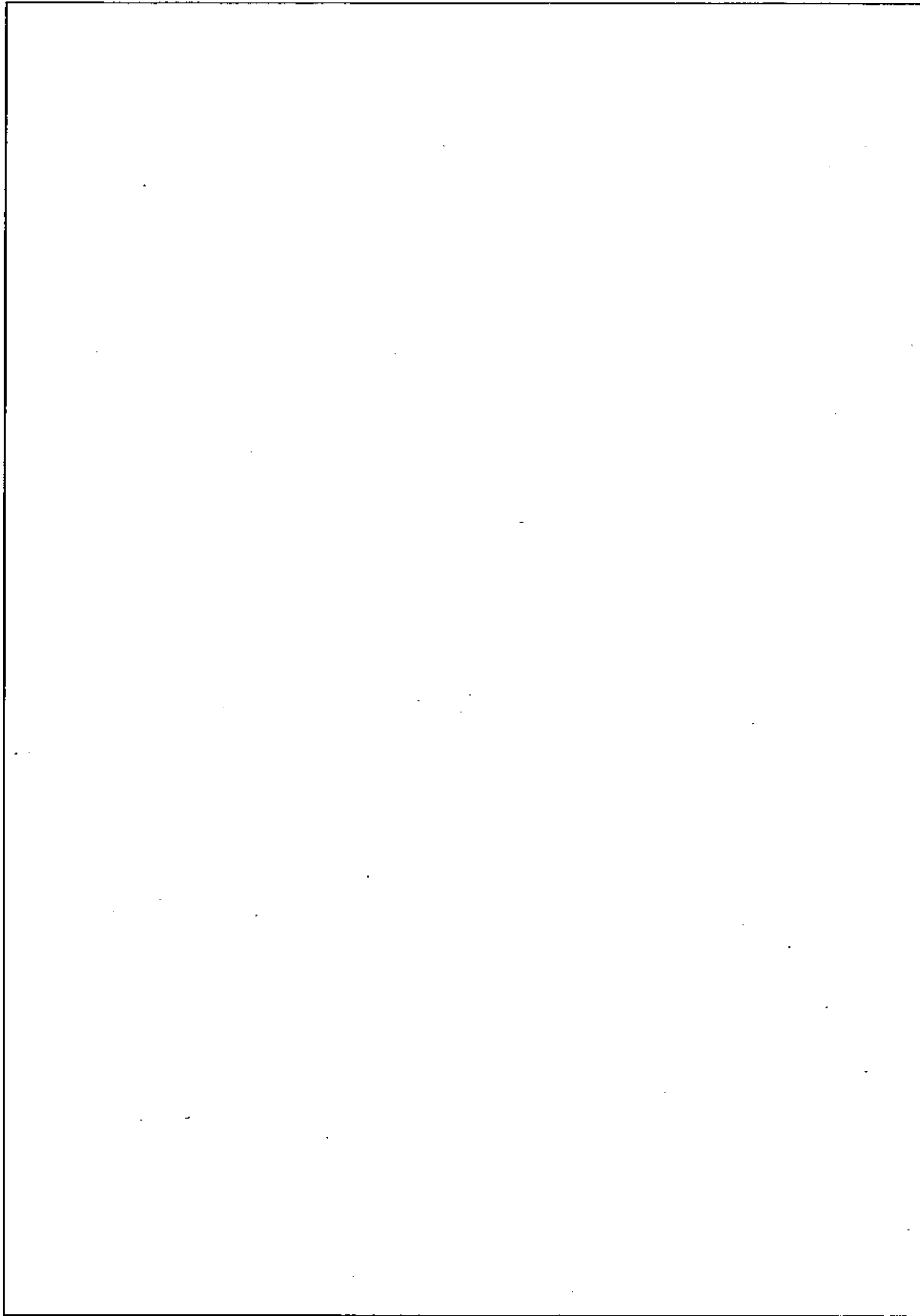
Continued

Answer :

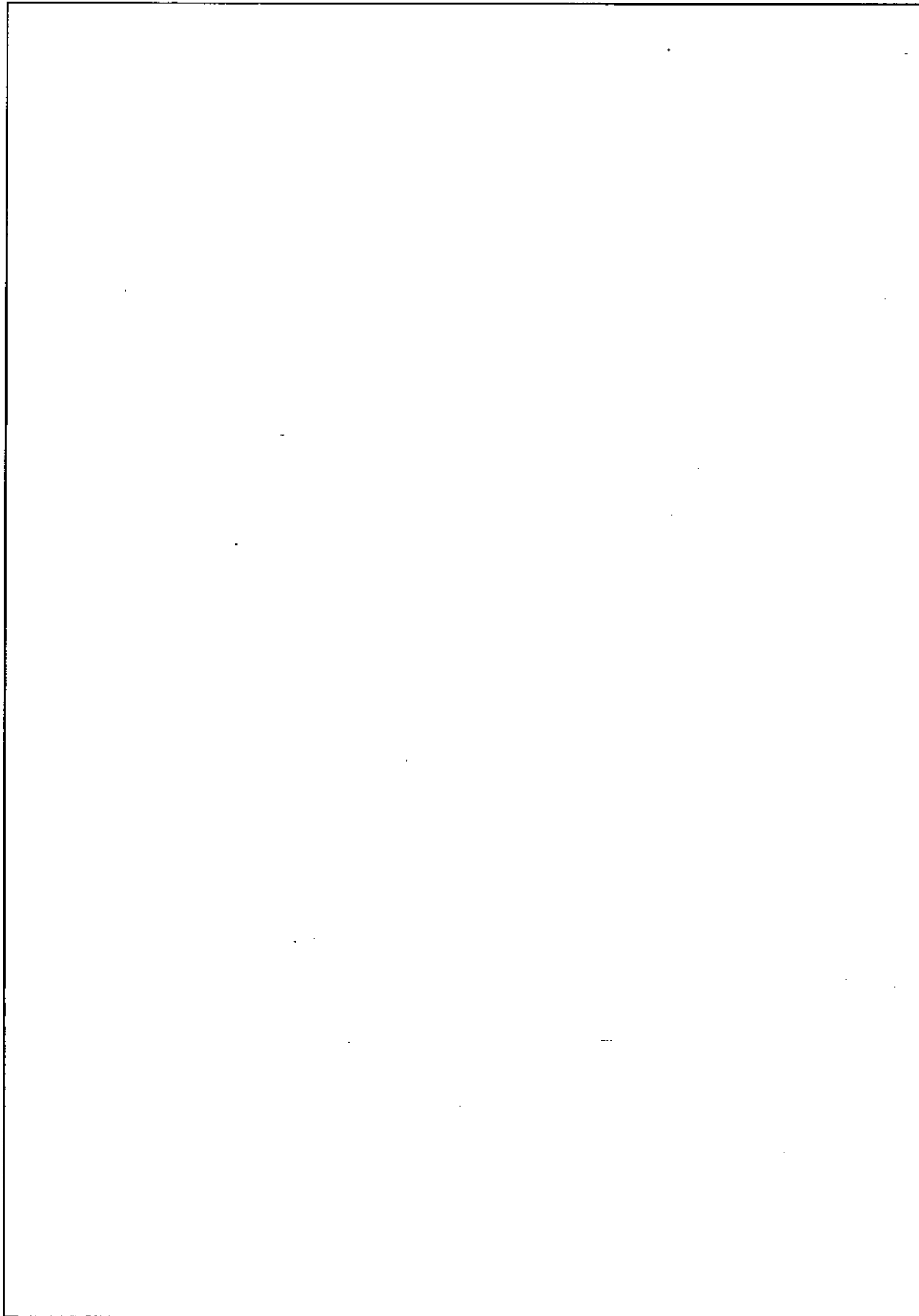
Continued



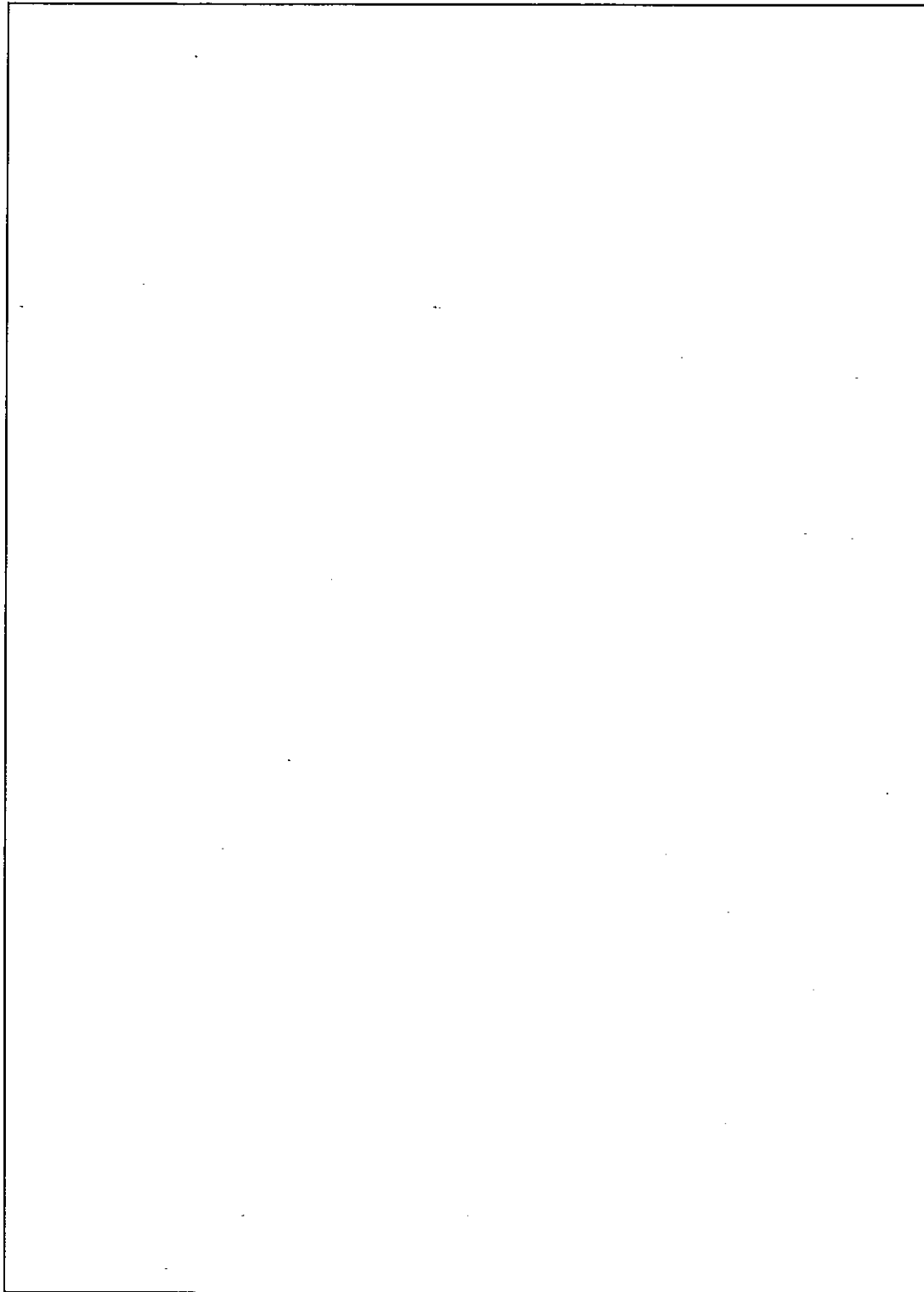
Continued



Continued



Continued



End of Page.